NaSPA technical SUPPORT

Supporting Enterprise Networks and Operating Environments

AUGUST 1995
VOLUME 3, NUMBER 8

MAXIMIZING YOUR AUTOMATION INVESTMENT

CLIENT-BASED ELECTRONIC REPORT DISTRIBUTION

AUTOMATION OF PROCESS AND CONFIGURATION MANAGEMENT

Simplifying Data Center Operations

NaSPA technical®
Supporting Enterprise Networks and Operating Environments
SUPPORT

AUGUST 1995
VOLUME 3, NUMBER 8

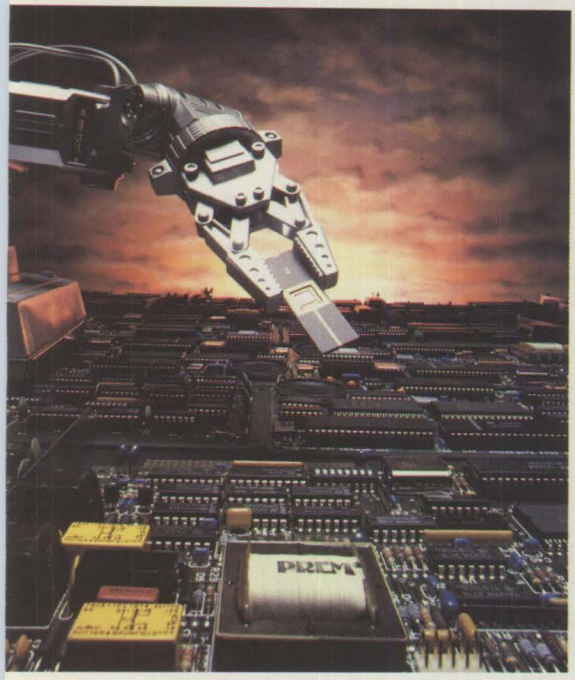


Photo: Don Carroll, Image Bank, Chicago

Automated operations simplifies the way data centers operate, enhancing productivity and system efficiency.

NaSPA Technical Support (ISSN 1079-3135) is published monthly by Technical Enterprises Inc., 4811 S. 76th St., Suite 210, Milwaukee, WI 53220-4362. Second-class postage paid at Milwaukee , WI and additional mailing office. POSTMASTER: Send address changes to ***NaSPA Technical Support***, 4811 S. 76th St., Suite 210,Milwaukee, WI 53220-4362.

FEATURES

STRATEGIES

ENTERPRISE

INSIGHTS

COMMUNICATIONS

COLUMNS

DEPARTMENTS

Dear NaSPA member;

ON THE 'NET

It is amazing the impact the Internet is having on the way we do business, shop, even educate our children. Estimated as having 20 million users, the Internet has permeated our society to create opportunities that we could only have imagined a few short years ago. In fact, the Internet has created a number of opportunities for you as a NaSPA member.

Last month we told you about NaSCOM's full Internet FTP and TELNET capabilities. Well I am pleased to report that NaSCOM usage has increased dramatically since we connected to the Internet. NaSCOM has been upgraded to include not only Internet access, but also RIP graphics, V.42bis support, and 12 online CD-ROMs.

If you haven't logged on to NaSCOM lately, now's your chance. Check out our World Wide Web page at http:\\www.NaSCOM.com! For more information, see the article in **NaSPA News** on page 7. Also, don't forget to check out the numerous product demos and storyboards available in DEMOS on DEMAND™ (see pages 40 and 41 for more details).

Stay tuned for additional coverage of the Internet in upcoming issues of ***Technical Support*** magazine. Our editors are working hard to provide you with informative articles that will help you navigate the 'net and make the most of your experiences. If you have any ideas for upcoming articles, please contact Editor Amy Birschbach at (414) 423-2420 Ext. 123 or Internet address editor@NaSCOM.com.

MEMBERSHIP HAS ITS REWARDS

NaSPA continually searches for and evaluates member benefits that truly help you professionally and personally. You will find a partial list of member benefits and services on pages 40 and 41 of this issue.

The list of member benefits expands this month with the addition of the NaSPA VISA card. In a special introductory offer, First Western Bank is offering zero percent interest until January 1, 1996! For more information, see their advertisement on page 14.

Speaking of advertisements, you may have noticed we have increased the size of ***Technical Support*** to 80 pages from 48. You are receiving more in-depth information on a variety of enterprisewide topics that are critical to you and your company. We salute the vendors whose advertisements appear in ***Technical Support***; it is their support that makes the page count grow. We encourage you to mention ***Technical Support*** when you contact these vendors.

Sincerely,

Scott Sherer

BPA audit applied for December 1994.

4811 S. 76th St., Suite 210
Milwaukee, WI 53220-4362
(414) 423-2420 FAX: (414) 423-2433
AUGUST 1995 VOLUME 3, NUMBER 8

PUBLISHING

Publisher
Scott Sherer. ext. 104
NaSCOM ID: SHERER
Internet ID: sherer@nascom.com

Editor, Vice President of Editorial Services:
Amy B. Birschbach, ext. 123
NaSCOM ID: EDITOR
CompuServe ID: 70373,1513
Internet ID: editor@nascom.com

Assistant Editor:
Matthew Ringlien, ext. 125
NaSCOM ID: EDIT1
Internet ID: edit1@nascom.com

Technical Editors:
Eric Allred, David Anderson, Mark Bell, Danal Estes, Elise Gotay, Israel E. Gotay, Mark Hanna, Howard Hauck, John E. Johnston, John D. Kinne, David Kreuter, Leo Langevin, Jim McMaster, Dwight S. Miller, Steve Pryor, Stephen L. Samson, Fred Schuff, Al Shing, Richard B. ViPond, Guy C. Yost

SALES/MARKETING

Vice President of Advertising Sales:
Jerry Seefeldt, ext. 110
NaSCOM ID: MARKET
Internet ID: market@nascom.com

East Coast Sales:
Cal Hart
(908) 495-6660

West Coast Sales:
Steve Cecil
(415) 595-2856

Sales Representative/List Manager:
Mike Czarnecki, ext. 105
NaSCOM ID:LISTMGR
Internet ID: listmgr@nascom.com

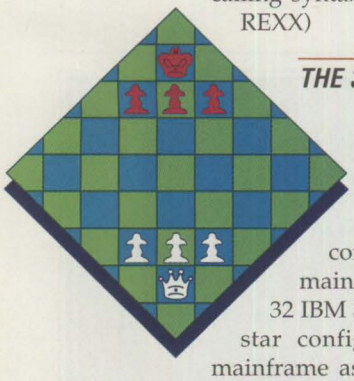
BY MARKUS PELT-LAYMAN

PDS Transfers Using EHLLAPI in REXX: Part I — An Overview of 3270 Concepts

With EHLLAPI, 3270 terminal emulation is supported over 3270 emulator cards, SDLC adapter cards or LAN adapter cards both under OS/2 as well as DOS/Windows.

The Emulator High-Level Language Application Programming Interface (EHLLAPI) running under OS/2 Communications Manager allows users to write client/server applications using emulated IBM 3270 terminals connected to IBM mainframes. Although EHLLAPI supports the Basic, C, COBOL, and Assembler languages, the power and flexibility of REXX significantly reduces programming effort and speeds up development.

EHLLAPI is available from almost every vendor that supports 3270 terminal emulation over 3270 emulator adapter cards, SDLC adapter cards or LAN adapter cards both under OS/2 as well as under DOS/Windows. (***Note:*** The RLIM.ZIP shareware package, available on the Rexx-R-Us BBS at (303) 440-1351 and on other BBSs provides EHLLAPI support specifically for Enterprise REXX and Quercus REXX under DOS and Windows. However, the calling syntax is slightly different than that for OS/2 REXX)



THE 3270 WORLD

The IBM 3270 family of terminals was already widely used before the advent of the personal computer. A typical local configuration (shown in Figure 1) consists of an IBM 3274 control unit (CU) connected via a channel to an IBM 370/390 mainframe. The control unit can connect up to 32 IBM 3270-type terminals via coaxial cable in a star configuration. Distances between CU and mainframe as well as between CU and terminal are limited without additional equipment.

For remote connections, a different control unit is used but the same 3270-type terminals can be attached to it again using coaxial cable in a star configuration. The control unit in this case is connected using modems and phone lines to a Front End Processor (FEP) which is attached via a channel to the mainframe.

Terminals come in two basic flavors: the older dumb Control Unit Terminals (CUT) which rely heavily on processing logic in the CU and the newer smart Distributed Function Terminals (DFT) which offload a lot of the terminal processing from the CU. 3270-compatible terminals come in many configurations and with many options such as APL and Katakana keyboards, security card readers, light pens, graphics, and IBM 328X printer support.

While the 3270 terminal and control unit market was very lucrative for both IBM and many compatible vendors, the introduction of the IBM PC saw the rise of 3270 emulation adapter cards. The initial problem for many large companies was that where previously every desk had a 3270 terminal on it, now each desk had a separate 3270 terminal and a PC (which didn't even talk to the mainframe). The solution was a 3270 emulation adapter card which allows a PC to act as a real 3270 terminal by attaching it via coax cable to a CU, eliminating the need for a standalone terminal.

While the adapter card provides the low-level hardware compatibility to talk to the control unit, the PC requires software to actually display output from the mainframe on the PC screen as well as accept keystrokes and send them to the mainframe. This emulation software is usually bundled with the adapter card and often is specific to the adapter card. Each vendor had its own way of doing things.

3270S IN A LAN WORLD

The introduction of LANs created yet another problem. PCs that were already connected to the mainframe using 3270 emulation adapter cards needed a LAN adapter card to connect to the LAN. In large corporations, this meant having each PC wired using two coaxial cables: one for the LAN and one for the mainframe. The solution to this double wiring problem was to connect to the mainframe over the LAN somehow, thus eliminating the 3270 emulation adapter card and coax cable. Of course, this introduced even more ways to provide 3270 emulation, again with each vendor doing things in non-standard ways.

EHLLAPI was the answer to standardize the way in which PC applications could programmatically address 3270 emulation terminals, regardless of the physical connection from PC to mainframe. Under OS/2 Communications Manager this is especially obvious, when you consider that OS/2 supports 3270 sessions using 3270 emulation adapter cards, Ethernet adapter cards, Token-Ring adapter cards, SDLC adapter cards, and even async serial adapter cards, as well as many others. Regardless of the hardware and software configuration, EHLLAPI provides the same interface to all.

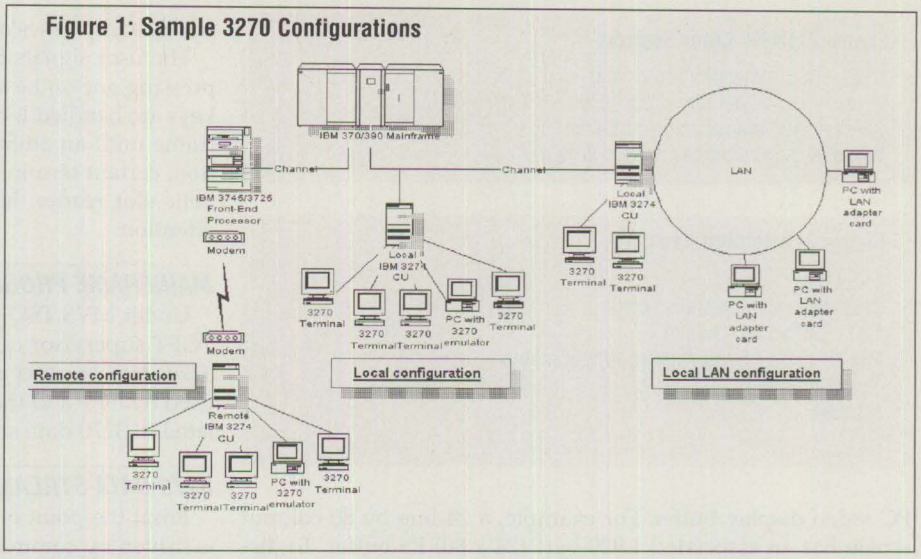
LU 6.2/APPC VS. 3270 EMULATION

Some people say that 3270 emulation is old technology and that it will go the way of the dinosaur. IBM is pushing LU 6.2 (a.k.a. APPC or Advanced Program-to-Program Communications) as the protocol of choice (the 3270 terminal protocol is referred to as LU2). While it is true that LU 6.2 is better designed and allows far more flexibility than 3270 terminal emulation, it does require that mainframe applications be rewritten to specifically use it. From a mainframe application point of view, it is still easier to write an application for a 3270 terminal than it is to write one for APPC. In addition, many legacy

systems would require major program logic changes to accommodate APPC.

This is partially due to the fact that existing 3270 terminal applications are encapsulated by a terminal monitor program such as CICS or TSO which is responsible for much of the VTAM network protocol establishment and security authentication processing. Until all mainframe applications are converted to use APPC, there will remain many legacy systems using 3270 terminals. 3270 terminal emulation may eliminate 3270 standalone terminals but not 3270 mainframe applications. What EHLLAPI can provide right now is a way to interface with such legacy systems and possibly even put a new coat of graphical user interface (GUI) paint on an old 3270 character-based mainframe application.

Figure 1: Sample 3270 Configurations



THE 3270 TERMINAL

The original 3270 terminal consisted of a green monochrome monitor and keyboard. The keyboard had a standard QWERTY layout with an additional 12 program function keys (PF keys) for special functions on the right hand side where the numeric keypad usually appears on PC keyboards.

The monitor displayed a character-based (as opposed to graphics) screen 24 lines by 80 columns, with an additional status line at the bottom of the screen (called the Operator Information Area or OIA) reserved for terminal session indicators (such as shift state, keyboard locked state and so forth). While the original 3270 models only supported limited display attributes for each character such as low- or high-intensity or invisible, the later models support different color and extended attributes such as reverse video and user-defined character sets (called programmed symbols in 3270 parlance) as well as different screen sizes.

FIELDS AND ATTRIBUTES

What is displayed on the screen is determined by what is placed in the terminal's presentation space (PS) buffer. In its simplest form the PS buffer acts like a

Figure 2: Basic Color Support

Red — high intensity input field
Green — low intensity input field
Blue — low intensity output field
White — high intensity output field

Figure 3: Attention Keys

Enter key
Clear key (also clears screen)
PA1, PA2 and PA3 keys
PF1 through PF24 program function keys
Test Req key
Attn key

PC video display buffer. For example, a 24-line by 80-column screen has an associated 1,920-byte (24 x 80) PS buffer. To display the character 'A' on line one, column one of the screen, place the byte 'A' in the first buffer location. The buffer address for the character at line two, column one is 80 (or offset 79).

Aside from straight text bytes, you can also place attribute bytes in the buffer. A buffer location set with an attribute byte marks the start of a new field. The value of the attribute byte describes some of the basic attributes for all characters that follow in the field up to the next attribute byte (the next field). Attribute bytes themselves display on screen as a blank (which cannot be modified via the keyboard).

For example, there is a bit in the definition of an attribute byte which indicates whether the field should be displayed in high-intensity or low-intensity. Another set of bits describes whether the field is an input field or output-only field (not modifiable by the keyboard user). One more bit is reserved to indicate whether the field has been modified (MDT or Modified Data Tag bit). This last bit is especially useful when reading the PS buffer as it indicates a field that has been changed in some way (Note: This bit can also be set by programs to simulate user modifications.)

The PS buffer wraps at the end of the screen: A field started by the last attribute byte will continue from line 24, column 80, to line one, column one, unless an attribute byte is put in line one, column one, or line 24, column 80.

EXTENDED ATTRIBUTES

To support more than the basic attributes described previously (intensity, input/output and modified), three additional buffers are used conceptually similar to the PS buffer in addressing. These extended attribute buffers are used for:

- extended highlighting (reverse video, blinking and underlined);
- extended colors (blue, red, pink, green, turquoise, yellow, and white); and
- programmed symbols (usually used for graphics).

Basic color support does not use extended attributes. A simple translation scheme using the basic field attributes is shown in Figure 2.

ATTENTION KEYS

The usual cycle for interaction on a 3270 terminal is:

- a mainframe program displays output on screen and waits for user input; and
- the user types information in input fields displayed on screen and presses an attention key.

The user signals that he is done typing input into fields by pressing one of the attention keys. All keys other than attention keys are handled locally and no data is transferred to the mainframe until an attention key is pressed. See Figure 3. In addition, certain terminal attachments such as a card reader, magnetic slot reader, hand scanner, or light pen can also signal attention.

MAINFRAME PROGRAMMING

Under MVS TSO you can use the TPUT (Terminal PUT) and TGET supervisor call instructions/macros to send and receive 3270 data streams respectively. At a higher level, the panels used by ISPF and the screen definitions used by CICS are translated to 3270 data streams.

3270 DATA STREAMS

From the point of view of the mainframe, the 3270 terminal is driven by commands. The basic commands are Write, Read buffer and Read Modified.

When a mainframe application sends the 3270 terminal a Write command it is followed by a sequence of data characters mixed with orders. There is a concept of current buffer address (or just buffer address for short) which is incremented whenever a data character is encountered in the write data stream (after the character is put into the PS buffer at the current buffer address). The orders encountered in a write stream can manipulate the PS buffer as well as alter the current buffer address. The changes made by the write stream to the PS buffer in turn change what is displayed on the screen of the terminal.

ORDERS

Figure 4 shows the various orders that can be included in a 3270 data stream and the effect each has on the PS buffer and current buffer address. When a Read Buffer command is sent by the mainframe to the 3270 terminal, the terminal transmits back to the host the entire PS buffer. The format of the returned data is data characters mixed with SF orders for each field attribute byte (so that the mainframe application can decipher where each field started). The Read Modified command works similarly but only returns data and SF orders for modified fields (fields with the MDT bit on in their basic attribute byte.) The MDT bit either was turned on by the user making modifications to the field using the keyboard or the mainframe application having sent the attribute byte originally with the MDT bit already on. The MDT bits are reset once all modified fields have been transmitted. The Read Modified command also adds SBA orders before every SF order to indicate where each modified field started.

The data stream returned by Read and Read Modified commands is preceded by three bytes which includes the AID (Attention Identification) character which identifies the key the user pressed to cause the Read or Read Modified to be completed.

PC PROGRAMMING — EHLLAPI

While 3270 data streams are used to access the 3270 terminal PS buffer from the mainframe, EHLLAPI is used to access the emulated terminal's PS buffer as well as send keystrokes from the PC side. The basic idea behind EHLLAPI is to send keystrokes and read the screen (i.e., access the PS buffer) to emulate what a human user would do manually. The EHLLAPI specification provides some 50 or more functions; luckily only a handful are needed for most applications. Before discussing the most often used EHLLAPI functions in detail, let's digress to the

problem that originally triggered the writing of this article.

PC/MAINFRAME FILE TRANSFER

Both MVS TSO and CMS include a command called IND$FILE which can be used to send or receive single files from a 3270 emulator to the mainframe and vice versa. The emphasis here is on the words SINGLE FILE (although the newest VM/CMS version of IND$FILE does allow multiple files using wildcard characters). Most terminal emulator products (including OS/2) provide a SEND and RECEIVE command on the PC side to initiate a file transfer to or from the mainframe, respectively. For example:

```
IND$FILE GET hostfile options
```

or

```
IND$FILE PUT hostfile options
```

The SEND and RECEIVE commands themselves issue a IND$FILE command to the mainframe host under the covers, so the user is never really aware of the command syntax of the IND$FILE host command. (If you want to check whether you have IND$FILE on your host, simply try the IND$FILE command without any operands from a 3270 session window. An indication that the command was not found probably means IND$FILE is not available.)

The syntax of the SEND and RECEIVE OS/2 commands is fully explained in the OS/2 online help (issue HELP SEND or HELP RECEIVE from an OS/2 command session). In brief, the syntax is:

Figure 4: 3270 Data Stream Orders

SBA — Set Buffer Address order changes the current buffer address to that specified in the order

RA — Repeat to Address order fills the PS buffer with the character specified in the order from the current buffer address to the one specified in the order.

SF — Start Field order inserts a basic attribute byte specified in the order into the PS buffer at the current buffer address. This starts a new field.

SFE — Start Field Extended order similar to SF order except the order may include both a basic attribute byte as well as extended attribute bytes.

MF — Modify Field order modifies basic and/or extended attributes for the field that starts at the current buffer address.

SA — Set Attribute order changes the extended attributes for a character at the current buffer address.

PT — Program Tab order changes the current buffer address to the next input field (starting from the current buffer address).

EAU — Erase All Unprotected order sets all input fields to nulls from the current buffer address to that specified in the order.

```
SEND pcfile [session:] hostfile
[options]
```

and

```
RECEIVE pcfile [session:] hostfile
[options]
```

where all lowercase items should be replaced with actual values and brackets indicate optional items (do not enter the brackets themselves). Check the OS/2 online help for the syntax details.

During the actual file transfer the session window usually goes blank. This is to prevent the user from becoming confused if he sees the file transfer data displayed (which may include non-viewable characters if binary data is transmitted).

PC vs. MAINFRAME FILES

PC files come in two basic flavors: ASCII text files and proprietary binary data files. Mainframe files are distinguished by file organization, record format, record length, and block size. During file transfer, differences in file structure and contents may have to be resolved. The major problems are:

- ASCII vs. EBCDIC encoding of text data. The PC uses ASCII encoding for text characters and the mainframe uses EBCDIC encoding. This means that whereas the letter "A" is represented in ASCII as a byte whose value is 65, in EBCDIC the letter is represented as a byte with the value 193. Obviously, without translation, an ASCII file will be printed or displayed as garbage on a mainframe.

Similarly, a mainframe file containing names and addresses will look like garbage when typed or printed on a PC without translation. Fortunately, the SEND and RECEIVE commands (and IND$FILE) can translate ASCII to EBCDIC and vice versa when you add the option 'ASCII' to the end of the command.

- CR/LF translation to records and vice versa. For text files on the PC, a line or record is marked by the presence of a carriage return and/or line feed character. On the mainframe, text files do not include these control characters. Instead, records in a file are all the same length (RECFM=3DF) or variable with a 4-byte record descriptor word preceding each record indicating the length of the record that follows. Clearly, when receiving a mainframe file without CR/LF control characters, the file is not readable on the PC. Conversely, the presence of CR/LF data will display as garbage on the mainframe and possibly cause problems with lines being split or running together erroneously.

Fortunately, the 'CRLF' option on the SEND and RECEIVE commands will translate records for text files correctly in this case.

- Record and block descriptor words for variable length mainframe files are *always* stripped when sent to the PC by IND$FILE. This basically means that variable length files are unusable unless they are text files and you use the 'CRLF' option.
- Little-Endian vs. Big-Endian problems can occur for binary files which include numeric values. The mainframe stores numeric values larger than a byte in most-significant-byte-first sequence. For example, the value 256 is stored in a file on the mainframe as two bytes '01'X and '00'X. On the PC, the least-significant bytes are always stored first (i.e., '00'X '01'X means the value 256).

What EHLLAPI can provide right now is a way to interface with such legacy systems and possibly even put a new coat of GUI paint on an old 3270 character-based mainframe application.

Unfortunately, there is no way to know which values would require switching bytes around without knowledge of a binary file's record layout and structure. IND$FILE will not provide any assistance for this problem. You may need to massage the file after transfer to make it usable on the target system in this case.

- File naming problems may occur. Since PC file names have a 11-character (8.3) format and mainframe Partitioned Data Set (PDS) member names have only eight characters, something will have to give. Usually, on the PC side the three-letter file name extension is an indication of file type, which on the mainframe is normally indicated by the last qualifier of the data set name. For example:

```
C:\MYPROJ\MYFILE.ASM
```

could be translated to:

```
'MYUSRID.MYPROJ.ASM(MYFILE)'
```

However, even with such a translation scheme, naming conventions for member names are stricter than for PC file names (which may include numerics as the first character, as well as special symbols not allowed for member names).

PARTITIONED DATA SETS

MVS supports a multitude of file organizations, one of which is the PDS. A PDS is a collection of homogeneous sequential files. All the member files (or members, for short) must share the same logical record length (LRECL), block size if blocked (BLKSIZE) and record format (RECFM=F for fixed-length records or RECFM=V for variable-length records). With the exception of the homogeneity requirement, PDSes are very similar to PC directories. While the mainframe has utilities to copy and manipulate entire PDSes and the PC has utilities to do the same for directories, the IND$FILE file transfer program has no direct support for PDS files.

To transfer an entire PDS, each member will need to be sent individually (i.e., a SEND or RECEIVE command must be issued for each member separately). This is a minor inconvenience for transferring a small number of members, but is a show-stopper if you have to transfer a PDS with hundreds of members. The solution, of course, is to write a small REXX program to issue the commands for us. Part II will examine in detail the design and coding of this REXX program. ts

Was this article of value to you? If so, please circle Reader Response Card No. 32.

Markus Pelt-Layman has been a software developer for more than 22 years. He is the author of several commercial software packages on IBM mainframes and PCs, including AF/Operator (sold by Candle Corp.) and co-author of the REXX interpreter in the OPS/MVS product (sold by Legent Corp.). He is currently working on RexxAnne, a REXX compiler for OS/2, Windows NT and Windows. He can be reached at Pelt Industries (800) 741-4322 or by email at markp@peltind.com.

A Picture is Worth a Thousand Words

BY JOHN E. JOHNSTON

This month I would like to call your attention to one of the many fine OS/2 shareware products on the market, JoeView. This product allows you to view graphic images using a Presentation Manager interface. The ability to display graphic images is becoming more and more important in the business world as document imaging systems and digital publishing become more established.

The author of JoeView built what he calls "exponential nagware" into the product. If you read the product's README file you will get the impression that he is tired of having people use his product without ever sending in the shareware registration. If you make extensive use of JoeView and do not register the product, you will witness the author's exponential nag-ware first hand. Don't worry, nothing malicious happens.

JOEVIEW FEATURES

Features of this shareware product include:

- Presentation Manager interface;
- fully-functioning graphic image display and manipulation utility;
- supports many graphic image formats, including: Targa (rle), PBM (ASCII), PBM (RAW), Sun Raster, PCX, Targa, JPEG, GIF, X11 BMP, TIFF, OS2 BMP, Windows BMP, and RLE Encode;
- allows you to create and display slide shows;
- allows you to manipulate images; and
- the ability to convert images from one format to another.

INSTALLATION

JoeView, like other OS/2 shareware programs, can be found in the OS2USER forum on CompuServe. Download file JVW122.ZIP from library 15. Unzip the file into a new directory on your hard drive, such as C:\JVW122. After unzipping, open an OS/2 session and change directories to the directory where JoeView resides. Enter JVWINSTL. This installation utility will ask you for a directory to install JoeView into. You can use the same directory that you unzipped JoeView into. When the install completes you will see the JoeView icon on your desktop.

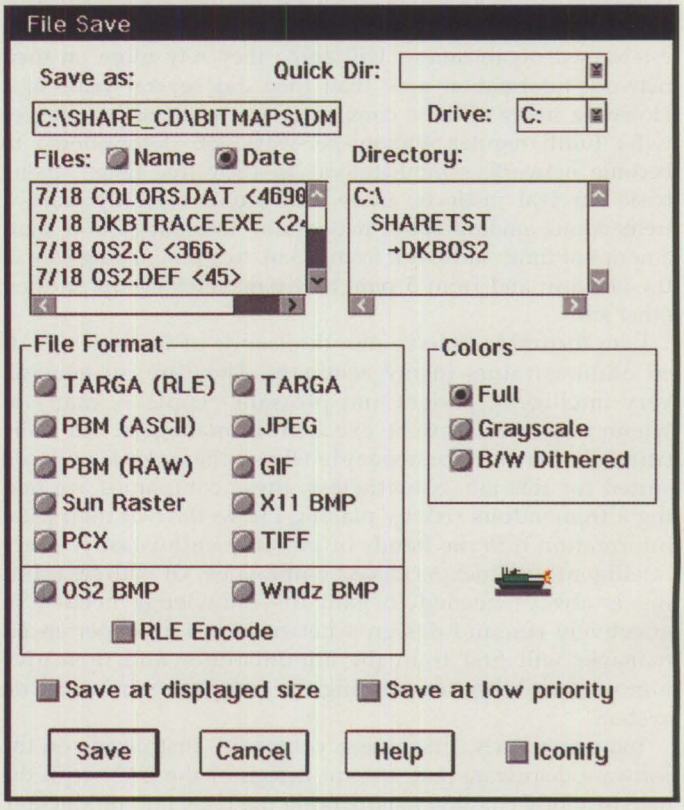
The first time you enter the application, either by typing joeview on the command line or by double clicking on the JoeView icon, a long initialization process will occur. This process can take anywhere from 20 minutes to an hour. A cute little graphic will be displayed, informing you of the progress of the initialization.

USAGE

When you double click on the JoeView icon, a graphic of one of the Bugs Bunny cartoon characters will be displayed. To access the functions of JoeView, move the cursor anywhere on this graphic and click the right mouse button. This will bring up a control dialog box.

To view a file, open the control dialog box as explained previously, click on Files, then click on Open. A standard open dialog box will be displayed from which you can navigate through your drives to select images. Double click on the desired image and JoeView will display it for you.

Figure 1: JoeView Version 1.22



JoeView provides several methods to manipulate your graphic images. From the control dialog box click on Manipulations. You can then select flip, rotate, invert colors, or zoom in.

You may also edit the images displayed by JoeView. From the control panel, select Edit. A submenu containing the following functions will be displayed:

- Copy and Paste;
- Colors;
- Resize;
- Crop;
- Auto Crop; and
- Smooth.

CREATING A SLIDE SHOW

You can use JoeView to create, save and play a slide show. From the control box, select Files, then select SlideShow. Use the drive and directory boxes to navigate through your hard drive(s) to find the graphic images that you wish to include in the show. As you select each directory, the files within that

directory will be displayed in the large box on the left-hand side of the screen. To add a slide to the show, highlight the filename by clicking on it with the left mouse button, then click on Add. You will see the name of the image displayed in the Selected Files box. After you have selected all of the images you wish to include in the slide show, you must either select a Timed or Manual presentation. Click on Options to set the time interval between slides. When you are finished, click on Save if you wish to save the slide show for future replays.

SAVING/CONVERTING AN IMAGE FILE

To save a file or convert a file to another file type, click on the Files menu item from the control box. Click on Save then change the "Save As" name and the File Format as shown in Figure 1, then click on Save.

FULL-FUNCTION, MULTI-PURPOSE

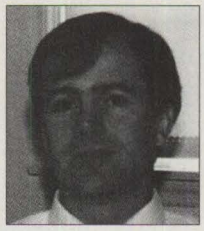
JoeView version 1.22 is a full-function, multi-purpose graphics viewing program. You can view and manipulate graphic images, change the file formats of images and develop and present slide shows. The controls of the program are intuitive and aren't the least bit clumsy.

JoeView is a full-function, multi-purpose graphics viewing program. You can view and manipulate graphic images, change the file formats of images and develop and present slide shows. JoeView is one of the best graphics viewing packages available under OS/2.

JoeView is one of the best graphics viewing packages available under OS/2.

If you have any questions, comments or ideas for future topics for this column, feel free to contact me via CompuServe address 73473,2146. ts

Was this column of value to you? If so, please circle Reader Response Card No. 49.



NaSPA member John E. Johnston is manager of technical support and communications for a major hospital in Pennsylvania. He designs and maintains cross-platform local and wide area networks utilizing NetWare, OS/2, DOS and Windows. John can be reached via NaSCOM ID Johnjohe or CompuServe ID 73473,2146.